

Date         

## ① Create New Project in Django

```
django-admin startproject New-Project
```

## ② Run newly created app:

```
python manage.py runserver
```

### \* Models:

- Models are Python classes that define the structure of your database.
- Use fields like CharField, IntegerField, ForeignKey etc.
- Example:

```
from django.db import models
class MyModel(models.Model):
    name = models.CharField(max-length=100)
    age = models.IntegerField()
```

### \* Views and Templates:

- views handle user requests and return appropriate responses.
- Templates are HTML files with placeholder for dynamic data.
- create a view in 'views.py' and a corresponding template in the 'template' directory.

## \* URL Patterns:

- Define URL patterns in the 'urls.py' file of your app.
- Map URLs to views using the 'urlpatterns' list.
- Example:

```
from django.urls import path
from . import views
```

```
urlpatterns = [
    path('my-url/', views.my-view, name='my-view'),
    ...
]
```

## \* Django Migrations:

- Migrations are a way to propagate changes you make to your models (adding a field, deleting a model) into your database schema.
- creating Migrations:
  - After defining models in model.py, create a migration using:

```
python manage.py makemigration
```

- Applying Migration:

- Apply the migration to database.

```
python manage.py migrate
```



- Creating a Superuser:

To access the Django admin interface, create a superuser account:

```
python manage.py createsuperuser
```

- Using Migrations in Development:

- During development, you might need to reset the database and apply migrations again.

```
python manage.py flush
```

- ★ Django Admin login:

1. Enabling Admin Interface:

- Ensure your app is included in the 'INSTALLED\_APPS' in 'settings.py'.
- Register models in the 'admin.py' file

```
from django.contrib import admin  
from .models import YourModel
```

```
admin.site.register(YourModel)
```

2. Superuser Login:

- Run the development server:

```
python manage.py runserver
```

- Access the admin interface at 'http://127.0.0.1:8000/admin/'.

- Log in with the superuser credentials created earlier.

### 3. customizing Admin interface:

- customize the admin interface by creating an admin.py file in each app.
- use 'list\_display', 'search\_fields' and other options to enhance the admin view.

# in app/admin.py

```
from django.contrib import admin
from .models import YourModel
```

```
@admin.register(YourModel)
```

```
class YourModelAdmin(admin.ModelAdmin):
    list_display = ('field1', 'field2')
    search_fields = ('field1')
```

### 4. Django Admin Commands:

Manage the admin interface using various commands, e.g. creating staff users.

```
"python manage.py createsuperuser --username=
Abc --email = abc@example.com --password =
secret"
```